1. Intro to Arrays

Lots of data. Lots of donuts. Mmm donuts.

Arrays hold multiple data values like a list.

1. Donuts to Code

An array is a data structure that you can use to store **multiple values** and arrays are also **organized**. Starts at 0.

1. Creating an Array

var donuts = [];

Keyboard shortcuts are powerful.

1. Accessing Array Elements

Index references the location, or **position**, of an element in an array.

1. Array Index
2. Quiz: UdaciFamily (6-1)

var udaciFamily = ["Julia", "James", "Brian"];

console.log(udaciFamily);

1. Quiz: Building the Crew (6-2)

var crew = [];

crew.push(captain);

crew.push(second);

crew.push(pilot);

crew.push(companion);

crew.push(mercenary);

crew.push(mechanic);

console.log(crew);

1. Quiz: The Price is Right (6-3)

prices[0] = 2.46;

prices[2] = 180.22;

prices[6] = 2.20;

console.log(prices);

1. Array Properties and Methods

Length, reverse, sort, push and pop.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>

Arrays are objects.

1. Length

console.log(donuts.length);

1. Push

You can use the push() method to add elements to the end of an array.

1. Pop

You can use the pop() method to remove elements from the end of an array.

1. Splice

splice() is another handy method that allows you to add and remove elements from anywhere within an array. The first argument represents the starting index from where you want to change the array, the second argument represents the numbers of elements you want to remove, and the remaining arguments represent the elements you want to add.

donuts.splice(1, 1, "chocolate cruller", "creme de leche"); // removes "chocolate frosted" at index 1 and adds "chocolate cruller" and "creme de leche" starting at index 1

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice>

1. Quiz: Colors of the Rainbow (6-4)

var rainbow = ["Red", "Orange", "Blackberry", "Blue"];

rainbow.splice(2, 1, "Yellow", "Green");

rainbow.splice(5, 0, "Purple");

console.log(rainbow); // [ 'Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Purple' ]

1. Quiz: Quidditch Cup (6-5)

function hasEnoughPlayers(teamArray) {

return teamArray.length > 6;

}

1. Quiz: Joining the Crew (6-6)

crew.push(doctor);

crew.push(sister);

crew.push(shepherd);

console.log(crew);

1. Quiz: Quiz: Checking out the Docs (6-7)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>

shift() will remove the first element from an array.

splice() can be used if you specify the index of the first element, and indicate that you want to delete 1 element.

Keep in mind that the slice() method allows you to create a copy of the array between two indices. Though you could just exclude the index of the first element, this approach does not directly modify a given array.

1. Array Loops

Once the data is in the array, you want to be able to efficiently access and manipulate each element in the array without writing repetitive code for each element.

for (var i = 0; i < donuts.length; i++) {// do something}

1. The forEach Loop

var donuts = ["jelly donut", "chocolate donut", "glazed donut"];

donuts.forEach(function(donut) {

donut += " hole";

donut = donut.toUpperCase();

console.log(donut);

});

The function that you pass to the forEach() method can take up to three parameters; these are called *element*, *index*, and *array*, but I can name them whatever I want.

1. Quiz: Another Type of Loop (6-8)

test.forEach(function(element, index, array) {

if (element % 3 === 0) {

array[index] += 100;

}

});

console.log(test);

1. Map

With the map() method, you can take an array, perform some operation on each element of the array, and return a new array. Map returns a new array, with the new values you calculated.

1. Quiz: I Got Bills (6-9)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number/toFixed>

var totals = bills.map(function(bill) {

bill = parseFloat((bill \* 1.15).toFixed(2));

return bill;

});

console.log(totals);

1. Arrays in Arrays

Imagine a grid, rows and columns.

1. 2D Donut Arrays

for (var row = 0; row < donutBox.length; row++) {

for (var column = 0; column < donutBox[row].length; column++) {

console.log(donutBox[row][column]);

}

}

1. Quiz: Nested Numbers (6-10)

for (var row = 0; row < numbers.length; row++) {

for (var column = 0; column < numbers[row].length; column++) {

if (numbers[row][column] % 2 === 0) {

numbers[row][column] = "even";

} else {

numbers[row][column] = "odd";

}

}

}

console.log(numbers);

1. Lesson 6 Summary

Objects in JS incoming!